

Please enter the substitute specification.

Please amend the claims as follows:

1. (Original) A system for generating a tree-based datastore comprising:
a processor;
a memory coupled to the processor; and
a tree-based datastore generator for creating at least one level of a tree-based datastore, the at least one level of the tree-based datastore comprising a first tree comprising a first root and at least one node of a plurality of nodes, a second tree comprising a second root and the at least one node of the first tree and at least a third tree comprising a third root and at least one of the plurality of nodes of the first tree.

Claims 2 – 35 (Cancelled)

36. (New) A method of evaluating a collection of data represented by an interlocking trees data store including nodes containing a count field and links between said nodes, said nodes including root nodes of which there are at least one primary root node and at least one elemental root node and which may include other root nodes, said nodes further including at least one end of thought node, at least one subcomponent node, and at least one end product node, and wherein there exist asResult and asCase links wherein said asResult links point between a root node and any other node, and wherein said asCase links point between at least one primary root node and at least one end product node and include in a path therebetween at least one subcomponent node, said method comprising the steps of:

determining a context within said data store and its corresponding value
determining a focus within said context and its corresponding value
calculating the probability of the occurrence of said focus within said context employing the corresponding values of said context and said focus.

37 (New) A method of evaluating a collection of data according to claim 36, wherein said step of determining a context and its corresponding value comprises the steps of:
selecting a context constraint list containing values represented by at least one root node, of said interlocking trees data store, wherein all of the at least one root nodes on said context constraint list are associated to each other by a logical expression;

identifying one or more paths by end product node, from the said at least one root node, by traversing from an asResult list of the at least one root node to the at least one root node's corresponding subcomponent node and then traversing asCase links between said corresponding subcomponent node to each corresponding end product node of said subcomponent node;

disregarding those paths that have links to elemental root nodes, the value fields of which do not conform with said logical expression, a resultant set of nodes thus forming a context being nodes along only those paths which have not been disregarded; and

adding the counts of the end product nodes of those one or more paths which have not been disregarded to obtain a context count.

38. (New) A method of evaluating a collection of data according to claim 37, wherein said logical expression includes at least one logical operator such as but not limited to, AND, OR, and NOT, GREATER THAN, LESS THAN, XNOR, EQUAL TO and any combination of such logical operators.

39. (New) A method of evaluating a collection of data according to claim 36, wherein said step of determining a context and its corresponding value comprises the steps of:

selecting a context constraint list containing values represented by at least one root node, of said interlocking trees data store, wherein all of the at least one root nodes on said context constraint list are associated to each other by a logical expression;

identifying one or more paths by end product node, by traversing from all possible end product nodes back toward the primary root using Case links along said path, and, at each subcomponent node using its Result link to locate and compare the root node to the said at least one root node;

disregarding those paths that have links to elemental root nodes, the value fields of which do not conform with said logical expression, a resultant set of nodes thus forming a context being nodes along only those paths which have not been disregarded; and

adding the counts of the end product nodes of those one or more paths, which have not been disregarded to obtain a context count.

40. (New) A method of evaluating a collection of data according to claim 39, wherein said logical expression includes at least one logical operator such as but not limited to, AND, OR, and NOT, GREATER THAN, LESS THAN, XNOR, EQUAL TO and any combination of such logical operators.

41. (New) A method of evaluating a collection of data according to claim 36, wherein said step of determining a context and its corresponding value comprises the steps of:
selecting all possible paths by end product node, of said interlocking trees data store,
disregarding those paths that have links to elemental root nodes, the value fields of which do not conform with said logical expression, a resultant set of nodes thus forming a context including nodes along only those paths which have not been disregarded; and
adding the counts of the end product nodes of those one or more paths which have not been disregarded to obtain a context count.

42. (New) A method of evaluating a collection of data according to claim 36, wherein said step of determining a focus and its corresponding value comprises the steps of:
selecting a focus constraint list of at least one root node, from the root nodes or the elemental root nodes, of said interlocking trees data store, said at least one root node being associated by a logical expression;
identifying one or more paths by end product node, from the said at least one root node, by traversing from the asResult list of the at least one root node to any corresponding subcomponent node and then traversing said corresponding subcomponent node's asCase links to its corresponding end product node.
disregarding those paths not within the established context and
also disregarding those paths that have links to elemental root nodes having value fields which do not conform to said logical expression, a resultant set of nodes thus forming a focus including nodes along only those paths which have not been disregarded, and
adding the counts of the end product nodes of those one or more paths which form said focus in order to obtain a focus count.

43. (New) A method of evaluating a collection of data according to claim 42, wherein said logical expression includes at least one logical operator such as but not limited to, AND, OR, and NOT, GREATER THAN, LESS THAN, XNOR, EQUAL TO and any combination of such logical operators.

44. (New) A method of evaluating a collection of data according to claim 36, wherein said step of determining a focus and its corresponding value comprises the steps of:

selecting a focus constraint list of at least one root node, from the root nodes or the elemental root nodes, of said interlocking trees data store, said at least one root node being associated by a logical expression;

identifying one or more paths by end product node, by traversing from all end product nodes within established context back along paths toward their primary root nodes, said paths identifiable using Case links of said end product nodes within established context, and while traversing, at each subcomponent node using the Result link to locate and compare the root node to the said at least one root node;

disregarding those paths that have links to elemental root nodes having value fields which do not conform to said logical expression, a resultant set of nodes thus forming a focus including nodes along only those paths which have not been disregarded; and,

adding the counts of the end product nodes of those one or more paths, which have not been disregarded to obtain a focus count.

45. (New) A method of evaluating a collection of data according to claim 44, wherein said logical expression includes at least one logical operator such as but not limited to, AND, OR, and NOT, GREATER THAN, LESS THAN, XNOR, EQUAL TO and any combination of such logical operators.

46. (New) A method of evaluating a collection of data represented by an interlocking trees data store including nodes containing a count field and links between said nodes, said nodes including root nodes of which there are at least one primary root node and at least one elemental root node and which may include other root nodes, said nodes further including at least one end of thought node, at least one subcomponent node, and at least one end product node, and wherein there exist asResult and asCase links wherein said asResult links point between a root node and any other node, and wherein said asCase links point between at least one primary root node and at least one end product node and include in a path therebetween at least one subcomponent node, said method comprising the steps of:

determining a context within said data set and its corresponding value

determining a position along each path of the context

determining a focus within said context and its corresponding value

calculating the probability of the occurrence of said focus between the said position and the end product, along the path within said context

47. (New) A method of evaluating a collection of data according to claim 46, wherein the step of determining a position along each path of the context comprises the steps of:
selecting a root node from the root nodes or the elemental root nodes, of said interlocking trees data store, and traversing from said root node's or elemental root node's asResult list to its corresponding subcomponent node in each path of the context.

48. (New) A method of evaluating a collection of data according to claim 46, wherein said step of determining a context and its corresponding value comprises the steps of:
selecting a context constraint list containing values represented by at least one root node, of said interlocking trees data store, wherein all of the at least one root nodes on said context constraint list are associated to each other by a logical expression;
identifying one or more paths by end product node, from the said at least one root node, by traversing from an asResult list of the at least one root node to the at least one root node's corresponding subcomponent node and then traversing asCase links between said corresponding subcomponent node to each corresponding end product node of said subcomponent node;
disregarding those paths that have links to elemental root nodes, the value fields of which do not conform with said logical expression, a resultant set of nodes thus forming a context being nodes along only those paths which have not been disregarded; and
adding the counts of the end product nodes of those one or more paths which have not been disregarded to obtain a context count.

49. (New) A method of evaluating a collection of data according to claim 48, wherein said logical expression includes at least one logical operator such as but not limited to, AND, OR, and NOT, GREATER THAN, LESS THAN, XNOR, EQUAL TO and any combination of such logical operators.

50. (New) A method of evaluating a collection of data according to claim 49, wherein said step of determining a context and its corresponding value comprises the steps of:
selecting a context constraint list containing values represented by at least one root node, of said interlocking trees data store, wherein all of the at least one root nodes on said context constraint list are associated to each other by a logical expression;
identifying one or more paths by end product node, by traversing from all possible end product nodes back toward the primary root using Case links along said path, and, at each

subcomponent node using its Result link to locate and compare the root node to the said at least one root node;

disregarding those paths that have links to elemental root nodes, the value fields of which do not conform with said logical expression, a resultant set of nodes thus forming a context being nodes along only those paths which have not been disregarded; and

adding the counts of the end product nodes of those one or more paths, which have not been disregarded to obtain a context count.

51. (New) A method of evaluating a calculating a collection of data according to claim 50, wherein said logical expression includes at least one logical operator such as but not limited to, AND, OR, and NOT, GREATERTHAN, LESSTHAN, XNOR, EQUALTO and any combination of such logical operators.

52. (New) A method of evaluating a collection of data according to claim 46, wherein said step of determining a context and its corresponding value comprises the steps of:
selecting all possible paths by end product node, of said interlocking trees data store,
disregarding those paths that have links to elemental root nodes, the value fields of which do not conform with said logical expression, a resultant set of nodes thus forming a context including nodes along only those paths which have not been disregarded; and
adding the counts of the end product nodes of those one or more paths which have not been disregarded to obtain a context count.

53. (New) A method of evaluating a collection of data according to claim 46, wherein said step of determining a focus and its corresponding value comprises the steps of:
selecting a focus constraint list of at least one root node, from the root nodes or the elemental root nodes, of said interlocking trees data store, said at least one root node being associated by a logical expression;
identifying one or more paths by end product node, from the said at least one root node, by traversing from the asResult list of the at least one root node to any corresponding subcomponent node and then traversing said corresponding subcomponent node's asCase links to its corresponding end product node.
disregarding those paths not within the established context and

also disregarding those paths that have links to elemental root nodes having value fields which do not conform to said logical expression, a resultant set of nodes thus forming a focus including nodes along only those paths which have not been disregarded, and adding the counts of the end product nodes of those one or more paths which form said focus in order to obtain a focus count.

54. (New) A method of evaluating a collection of data according to claim 53, wherein said logical expression includes at least one logical operator such as but not limited to, AND, OR, and NOT, GREATERTHAN, LESSTHAN, XNOR, EQUALTO and any combination of such logical operators.

55. (New) A method of evaluating a collection of data according to claim 46, wherein said step of determining a focus and its corresponding value comprises the steps of:

selecting a focus constraint list of at least one root node, from the root nodes or the elemental root nodes, of said interlocking trees data store, said at least one root node being associated by a logical expression;

identifying one or more paths by end product node, by traversing from all end product nodes within established context back along paths toward their primary root nodes, said paths identifiable using Case links of said end product nodes within established context, and while traversing, at each subcomponent node using the Result link to locate and compare the root node to the said at least one root node;

disregarding those paths that have links to elemental root nodes having value fields which do not conform to said logical expression, a resultant set of nodes thus forming a focus including nodes along only those paths which have not been disregarded; and, adding the counts of the end product nodes of those one or more paths, which have not been disregarded to obtain a focus count.

56. (New) A method of evaluating a collection of data according to claim 55, wherein said logical expression includes at least one logical operator such as but not limited to, AND, OR, and NOT, GREATERTHAN, LESSTHAN, XNOR, EQUALTO and any combination of such logical operators.

57. (New) A method of evaluating a collection of data represented by an interlocking trees data store including nodes containing a count field and links between said nodes, said nodes

including root nodes of which there are at least one primary root node and at least one elemental root node and which may include other root nodes, said nodes further including at least one end of thought node, at least one subcomponent node, and at least one end product node, and wherein there exist asResult and asCase links wherein said asResult links point between a root node and any other node, and wherein said asCase links point between at least one primary root node and at least one end product node and include in a path therebetween at least one subcomponent node, said method comprising the steps of:

- determining a context within said data set and its corresponding value
- determining a position along each path of the context
- determining a focus within said context and its corresponding value
- calculating the probability of the occurrence of said focus between the said position and the primary root, along the path within said context

58. (New) A method of evaluating a collection of data according to claim 57, wherein the step of determining a position along each path of the context comprises the steps of:

- selecting a root node from the root nodes or the elemental root nodes, of said interlocking trees data store, and traversing from said root node's or elemental root node's asResult list to its corresponding subcomponent node in each path of the context.

59. (New) A method of evaluating a collection of data according to claim 57, wherein said step of determining a context and its corresponding value comprises the steps of:

- selecting a context constraint list containing values represented by at least one root node, of said interlocking trees data store, wherein all of the at least one root nodes on said context constraint list are associated to each other by a logical expression;
- identifying one or more paths by end product node, from the said at least one root node, by traversing from an asResult list of the at least one root node to the at least one root node's corresponding subcomponent node and then traversing asCase links between said corresponding subcomponent node to each corresponding end product node of said subcomponent node;
- disregarding those paths that have links to elemental root nodes, the value fields of which do not conform with said logical expression, a resultant set of nodes thus forming a context being nodes along only those paths which have not been disregarded; and
- adding the counts of the end product nodes of those one or more paths which have not been disregarded to obtain a context count.

60. (New) A method of evaluating a collection of data according to claim 59, wherein said logical expression includes at least one logical operator such as but not limited to, AND, OR, and NOT, GREATERTHAN, LESSTHAN, XNOR, EQUALTO and any combination of such logical operators.

61. (New) A method of evaluating a collection of data according to claim 57, wherein said step of determining a context and its corresponding value comprises the steps of:

- selecting a context constraint list containing values represented by at least one root node, of said interlocking trees data store, wherein all of the at least one root nodes on said context constraint list are associated to each other by a logical expression;
- identifying one or more paths by end product node, by traversing from all possible end product nodes back toward the primary root using Case links along said path, and, at each subcomponent node using its Result link to locate and compare the root node to the said at least one root node;
- disregarding those paths that have links to elemental root nodes, the value fields of which do not conform with said logical expression, a resultant set of nodes thus forming a context being nodes along only those paths which have not been disregarded; and
- adding the counts of the end product nodes of those one or more paths, which have not been disregarded to obtain a context count.

62. (New) A method of evaluating a collection of data according to claim 61, wherein said logical expression includes at least one logical operator such as but not limited to, AND, OR, and NOT, GREATERTHAN, LESSTHAN, XNOR, EQUALTO and any combination of such logical operators.

63. (New) A method of evaluating a collection of data according to claim 57, wherein said step of determining a context and its corresponding value comprises the steps of:

- selecting all possible paths by end product node, of said interlocking trees data store, disregarding those paths that have links to elemental root nodes, the value fields of which do not conform with said logical expression, a resultant set of nodes thus forming a context including nodes along only those paths which have not been disregarded; and
- adding the counts of the end product nodes of those one or more paths which have not been disregarded to obtain a context count.

64. (New) A method of evaluating a collection of data according to claim 57, wherein said step of determining a focus and its corresponding value comprises the steps of:

selecting a focus constraint list of at least one root node, from the root nodes or the elemental root nodes, of said interlocking trees data store, said at least one root node being associated by a logical expression;

identifying one or more paths by end product node, from the said at least one root node, by traversing from the asResult list of the at least one root node to any corresponding subcomponent node and then traversing said corresponding subcomponent node's asCase links to its corresponding end product node.

disregarding those paths not within the established context and

also disregarding those paths that have links to elemental root nodes having value fields which do not conform to said logical expression, a resultant set of nodes thus forming a focus including nodes along only those paths which have not been disregarded, and adding the counts of the end product nodes of those one or more paths which form said focus in order to obtain a focus count.

65. (New) A method of evaluating a collection of data according to claim 64, wherein said logical expression includes at least one logical operator such as but not limited to, AND, OR, and NOT, GREATER THAN, LESS THAN, XNOR, EQUAL TO and any combination of such logical operators.

66. (New) A method of evaluating a collection of data according to claim 57, wherein said step of determining a focus and its corresponding value comprises the steps of:

selecting a focus constraint list of at least one root node, from the root nodes or the elemental root nodes, of said interlocking trees data store, said at least one root node being associated by a logical expression;

identifying one or more paths by end product node, by traversing from all end product nodes within established context back along paths toward their primary root nodes, said paths identifiable using Case links of said end product nodes within established context, and while traversing, at each subcomponent node using the Result link to locate and compare the root node to the said at least one root node;

disregarding those paths that have links to elemental root nodes having value fields which do not conform to said logical expression, a resultant set of nodes thus forming a focus including nodes along only those paths which have not been disregarded; and,

adding the counts of the end product nodes of those one or more paths, which have not been disregarded to obtain a focus count.

67. (New) A method of evaluating a collection of data according to claim 66, wherein said logical expression includes at least one logical operator such as but not limited to, AND, OR, and NOT, GREATER THAN, LESS THAN, XNOR, EQUAL TO and any combination of such logical operators.

68. (New) A structure comprising nodes and links between said nodes, said nodes having a plurality of data fields, at least two of said plurality of data fields containing a pointer, one of said at least two pointers being a Case pointer and the other of said at least two pointers being a Result pointer and at least one node having at least one additional pointer to a list of pointers, one of said additional pointers to said list of pointers being to an asCase list in instances where said node has associated asCase list and another being to asResult list in instances where said node has associated an asResult list, and wherein said nodes contain a count field, and wherein said nodes include root nodes of which there are at least one primary root node and at least one elemental root node and wherein said nodes may include other root nodes, said nodes further including at least one end of thought node, at least one subcomponent node, and at least one end product node, and wherein said asResult links point between a root node and any other node, and wherein said asCase links point between at least one primary root node and at least one end product node include in a path therebetween containing at least one subcomponent node and wherein said asResult links point between a root or end product node and a subcomponent node or end product node on said path therebetween, and wherein said elemental nodes also have a field having a value.

69. (New) The structure of claim 68 wherein said structure is formed from a set of program instructions which configure a computer system when activated therein to produce said structure.

70. (New) A computer readable medium containing the set of program instructions as set forth in claim 69.

71. (New) The structure set forth in claim 68 wherein said count field contains an intensity variable, said intensity variable modifiable at various intensities corresponding to various predetermined traversal types of activity related to a node containing said count field.

72. (New) structure as set forth in claim 68 wherein said asCase and said asResult lists are stored in a separate data structure from said interlocking trees structure and wherein said separate data structure is associated with related nodes in said interlocking trees structure by pointers.

73. (New) A structure comprising nodes and links between said nodes, said nodes having a plurality of data fields, at least two of said plurality of data fields containing a pointer, one of said at least two pointers being a Case pointer and the other of said at least two pointers being a Result pointer and at least one node having at least one additional pointer to a list of pointers, one of said additional pointers to said list of pointers being to an asCase list in instances where said node has associated asCase list and another being to asResult list in instances where said node has associated an asResult list, and wherein said nodes are provided with one sub-node for each predetermined manner of traversal, said sub-nodes containing a count field for recording traversals of said nodes in predetermined manners, and wherein said nodes include root nodes of which there are at least one primary root node and at least one elemental root node and wherein said nodes may include other root nodes, said nodes further including at least one end of thought node, at least one subcomponent node, and at least one end product node, and wherein said asResult links point between a root node and any other node, and wherein said asCase links point between at least one primary root node and at least one end product node include in a path therebetween containing at least one subcomponent node and wherein said asResult links point between a root or end product node and a subcomponent node or end product node on said path therebetween, and wherein said elemental nodes also have a field having a value.

74. (New) A structure comprising nodes and links between said nodes, said nodes having a plurality of data fields, at least two of said plurality of data fields containing a pointer, one of said at least two pointers being a Case pointer and the other of said at least two pointers being a Result pointer and at least one node having at least one additional pointer to a list of pointers, one of said additional pointers to said list of pointers being to an asCase list in instances where said node has associated asCase list and another being to asResult list in instances where said node has associated an asResult list, and wherein said nodes contain an additional field, and

wherein said nodes include root nodes of which there are at least one primary root node and at least one elemental root node and wherein said nodes may include other root nodes, said nodes further including at least one end of thought node, at least one subcomponent node, and at least one end product node, and wherein said asResult links point between a root node and any other node, and wherein said asCase links point between at least one primary root node and at least one end product node include in a path therebetween containing at least one subcomponent node and wherein said asResult links point between a root or end product node and a subcomponent node or end product node on said path therebetween, and wherein said elemental nodes also have a field having a value.

75. (New) The structure of claim 74 wherein said additional field is a count field.